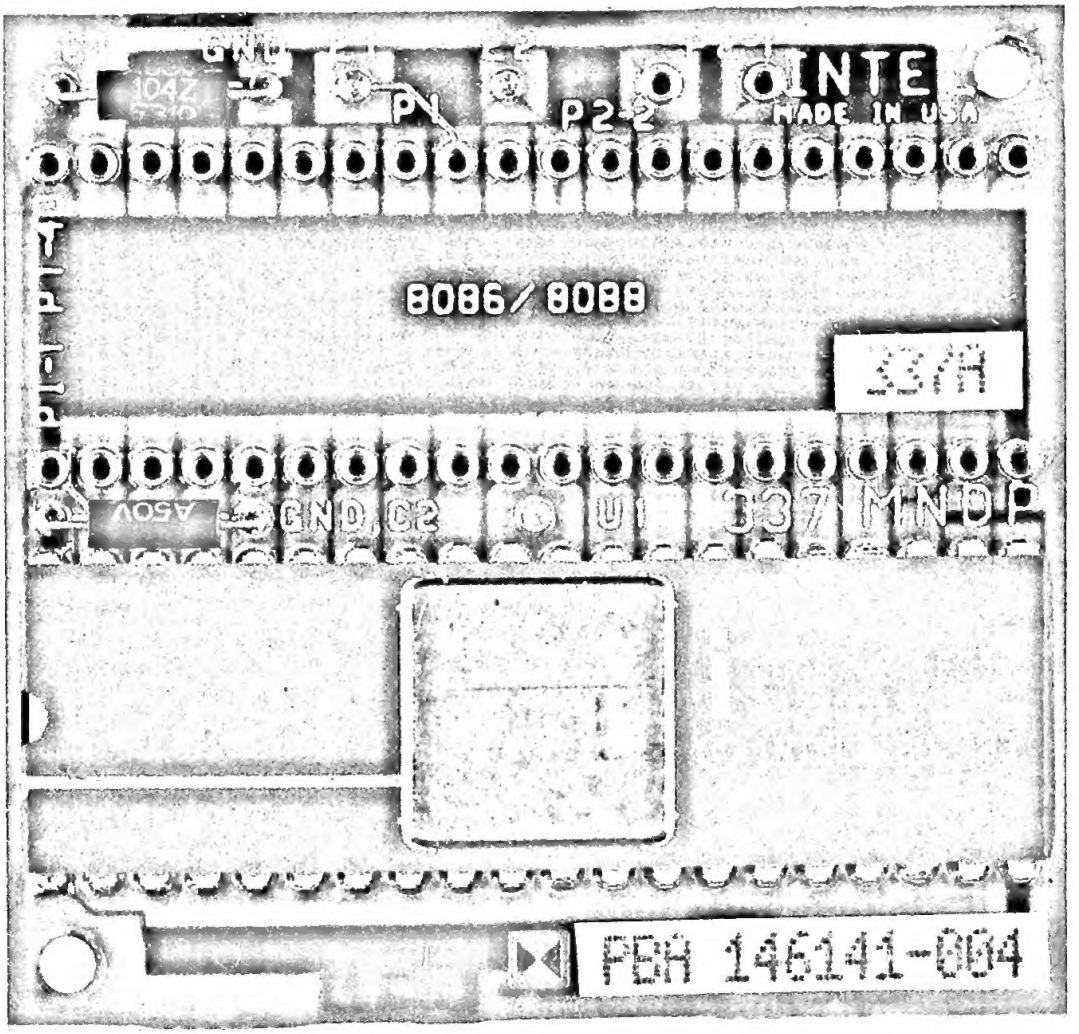
intel

iSBC® 337A MULTIMODULE™ NUMERIC DATA PROCESSOR HARDWARE REFERENCE MANUAL



Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

Order Number: 147163-001

iSBC® 337A MULTIMODULE™ NUMERIC DATA PROCESSOR HARDWARE REFERENCE MANUAL

Order Number: 147163-001

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

BITBUS	iLBX	iPDS	OpenNET
COMMputer	im	iRMX	Plug-A-Bubble
CREDIT	iMDDX	iSBC	PROMPT
Data Pipeline	iMMX	iSBX	Promware
Genius	Insite	iSDM	QueX
i	Intel	iSXM	QUEST
i2ICE	intel	Library Manager	Ripplemode
ICE	intelBOS	MCS	RMX/80
iCS	Intelevision	Megachassis	RUPI
iDBP	inteligent Identifier	MICROMAINFRAME	Seamless
iDIS	inteligent Programming	MULTIBUS	SLD
	Intellec	MULTICHANNEL	SYSTEM 2000
	Intellink	MULTIMODULE	UPI
	iOSP		

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

*MULTIBUS is a patented Intel bus.

REV.	REVISION HISTORY	DATE
-001	Original Issue.	11/84

PREFACE

This manual provides general information, installation instructions, programming information, principles of operation, and service information for the iSBC 337A MULTIMODULE Numeric Data Processor board. Additional information is available in the following document:

- Intel Microsystem Components Handbook, Order No: 23Ø843.

***

CONTENTS

PAGE

CHAPTER 1
GENERAL INFORMATION
1.1 Introduction..... 1-1
1.2 Description..... 1-1
1.3 Equipment Supplied..... 1-1
1.4 Specifications..... 1-2

CHAPTER 2
PREPARATION FOR USE
2.1 Introduction..... 2-1
2.2 Unpacking and Inspection..... 2-1
2.3 Installation Considerations..... 2-1
2.3.1 Power Requirements..... 2-2
2.3.2 Cooling Requirements..... 2-2
2.3.3 Physical Dimensions..... 2-2
2.3.4 Connector Configuration..... 2-2
2.3.5 Jumper Configuration..... 2-2
2.3.6 Installation Procedure..... 2-2

CHAPTER 3
PROGRAMMING INFORMATION
3.1 Introduction..... 3-1
3.2 Programming Interface..... 3-1
3.3 Instruction Set..... 3-2
3.4 Data Formats..... 3-7
3.4.1 Real..... 3-7
3.4.2 Integer..... 3-10
3.4.3 Packed Binary Coded Decimal..... 3-11
3.5 Status Format..... 3-11
3.5.1 Control Word..... 3-12
3.5.2 Status Word..... 3-14
3.5.3 Tag Word..... 3-17
3.6 Example Problems for Floating Point..... 3-17
3.6.1 Problem Two Program Detail..... 3-19

CHAPTER 4
PRINCIPLES OF OPERATION
4.1 Introduction..... 4-1
4.2 The CPU/NDP Relationship..... 4-1
4.2.1 Bus Operations..... 4-4
4.2.2 Read Operation..... 4-4
4.2.3 Write Operation..... 4-5
4.3 8087-2 Architecture..... 4-5
4.3.1 Control Unit..... 4-5

PAGE

CHAPTER 4 (continued)
4.3.2 Numeric Execution Unit..... 4-7
4.3.3 8Ø87-2 Internal File..... 4-7
4.3.4 Deadlock..... 4-7

CHAPTER 5
SERVICE INFORMATION
5.1 Introduction..... 5-1
5.2 Replaceable Parts..... 5-1
5.3 Service Diagrams..... 5-1
5.4 Service and Repair Assistance..... 5-1

TABLES

1-1. Specifications 1-2

2-1. Connectors P1 and P2 Pin Assignments..... 2-4
2-2. Connectors P1 and P2 Signal Functions..... 2-5

3-1. 8Ø87-2 NDP Instructions..... 3-2
3-2. Condition Code Settings..... 3-15
3-3. Stack Image Per Stack..... 3-2Ø

5-1. Replaceable Parts..... 5-3
5-2. List of Manufacturers' Codes..... 5-3

FIGURES

PAGE

2-1. Board Dimensions (Inches)..... 2-3
2-2. Mounting Clearances (Inches)..... 2-4

3-1. Status Information Format..... 3-12
3-2. Overall Programming Example Flow Chart..... 3-2Ø
3-3. Detailed Programming Example Flow Chart..... 3-21

4-1. iSBC® 337A MNDP System Configuration..... 4-2
4-2. Request/Grant Sequence Timing..... 4-3
4-3. 8Ø87-2 Block Diagram..... 4-6
4-4. 8Ø87-2 Internal File Format..... 4-8

5-1. iSBC® 337A Board Parts Location Diagram..... 5-4
5-2. iSBC® 337A Board Schematic..... 5-5

***

CHAPTER 1
GENERAL INFORMATION

1.1 INTRODUCTION

The iSBC 337A MULTIMODULE Numeric Data Processor (MNDP) board is a co-processor that performs arithmetic and comparison operations on a variety of numeric data types. It also executes numerous built-in trigonometric and transcendental functions (e.g., tangent and log functions). The iSBC 337A MNDP option will operate with all iSBC 86 and iSBC 88 based single board computers. The iSBC 337A MNDP effectively extends the register and instruction sets of the host CPU on these boards and adds several new data types as well.

1.2 DESCRIPTION

The iSBC 337A board is based on the 8Ø87-2 Numeric Data Processor. The 8Ø87-2 provides powerful arithmetic operations on seven different data types: single and double precision floating point numbers, words, short and long integers, BCD format, and internal file precision.

The iSBC 337A board, which is a plug-on option, operates concurrently with the 8Ø86/8Ø88 CPU. The iSBC 337A board can also be utilized by other 8Ø86/8Ø88 based designs, due to the unique characteristics of the co-processor and MULTIMODULE interface.

1.3 EQUIPMENT SUPPLIED

The following is supplied with the iSBC 337A board.

Schematic Diagram, drawing number: 142698.

1.4 SPECIFICATIONS

Specifications for the iSBC 337A board are listed in Table 1-1.

Table 1-1. Specifications

Physical Characteristics	
Width	5.Ø8 cm (2.ØØ inches)
Length	5.334 cm (2.1Ø inches)
Height	Ø.594 cm (Ø.234 inches) iSBC 337A board only 1.82 cm (Ø.718 inches) iSBC 337A board and iSBC board
Weight	17.33 gm (Ø.576 oz.)
Environmental Requirements	
Operating Temperature	Ø° to 55°C (32° to 131°F)
Relative Humidity	To 9Ø% without condensation
Power Requirements (Maximum)	
V_{CC} = +5V ±5%	
I_{CC} = 475 mA	

***

CHAPTER 2
PREPARATION FOR USE

2.1 INTRODUCTION

This chapter provides information for preparing and installing the iSBC 337A MULTIMODULE NDP board. Important information on unpacking and inspection, installation considerations, the connector configuration, and installation procedure are included.

2.2 UNPACKING AND INSPECTION

Inspect the shipping carton immediately upon receipt for evidence of mishandling during transit. If the shipping carton is severely damaged or water-stained, request that the carrier's agent be present when the carton is opened. If the carrier's agent is not present when the carton is opened and the contents of the carton are damaged, keep the carton and packing material for the agent's inspection.

For repairs to a product damaged in shipment, contact the Intel MCSD Technical Support Center to obtain a return authorization number and further instructions. A purchase order will be required to complete the repair. A copy of the purchase order should be submitted to the carrier with your claim.

It is suggested that salvageable shipping cartons and packing material be saved for future use in the event the product must be transported.

2.3 INSTALLATION CONSIDERATIONS

The iSBC 337A board is designed to mount on an iSBC 86 or iSBC 88 single board computer. It can also be used on other board designs utilizing the 8Ø86 or 8Ø88 CPU. Installation considerations such as power, cooling, mounting, and physical size requirements are outlined in the following sections.

2.3.1 POWER REQUIREMENTS

The iSBC 337A board requires +5V (±Ø.25V) at 475 mA maximum, supplied by the host board through the mounting connector.

2.3.2 COOLING REQUIREMENTS

The iSBC 337A board dissipates 35.3 gram-calories/minute (Ø.143 BTU/minute) and adequate circulation of air must be provided to prevent a temperature rise above 55°C (131°F).

2.3.3 PHYSICAL DIMENSIONS

The physical dimensions are listed in Table 1-1.

Figure 2-1 shows the physical dimensions and Figure 2-2 shows clearances for an iSBC 337A board mounted on a host iSBC microcomputer. The dimensions shown in Figure 2-2 are maximum heights.

2.3.4 CONNECTOR CONFIGURATION

Connectors P1 and P2 interface all input and output signals on the iSBC 337A board. The signals on each pin of the P1 and P2 connectors are listed in Table 2-1 and descriptions of the signal functions are listed in Table 2-2.

2.3.5 JUMPER CONFIGURATION

There is only one jumper on the iSBC 337A board, between E1 and E2, and is not used.

2.3.6 INSTALLATION PROCEDURE

The iSBC 337A board is designed to be mounted in the existing 8Ø86/8Ø88 CPU IC socket. The following steps describe the method of installing the iSBC 337A board on any Intel iSBC 86 or iSBC 88 single board computer.

1. Remove the 8Ø86/8Ø88 CPU from the base board.

2. Insert the iSBC 337A board mating pins into the CPU's IC socket and the other mating pin (P2) into the corresponding jack on the base board. (The mating of P2 and this jack connects the interrupt output of the 8Ø87-2 MNDP to the 8259A interrupt array on the base board.)

3. After ensuring that the seating is firm, insert the previously removed processor into the CPU socket of the iSBC 337A board (refer to Figure 2-1).

4. An optional wire may be soldered between the iSBC 337A board and the host board for security. The solder pad is located between the two IC's mounted on the iSBC 337A board. A matching solder pad is located on the host board directly below the solder pad on the iSBC 337A board when it is mounted.

5. The MINT signal (pin 1) is connected to a matching receptacle on the base board on all Intel iSBC 86 or 88 based products. This receptacle supplies a path to the interrupt matrix.

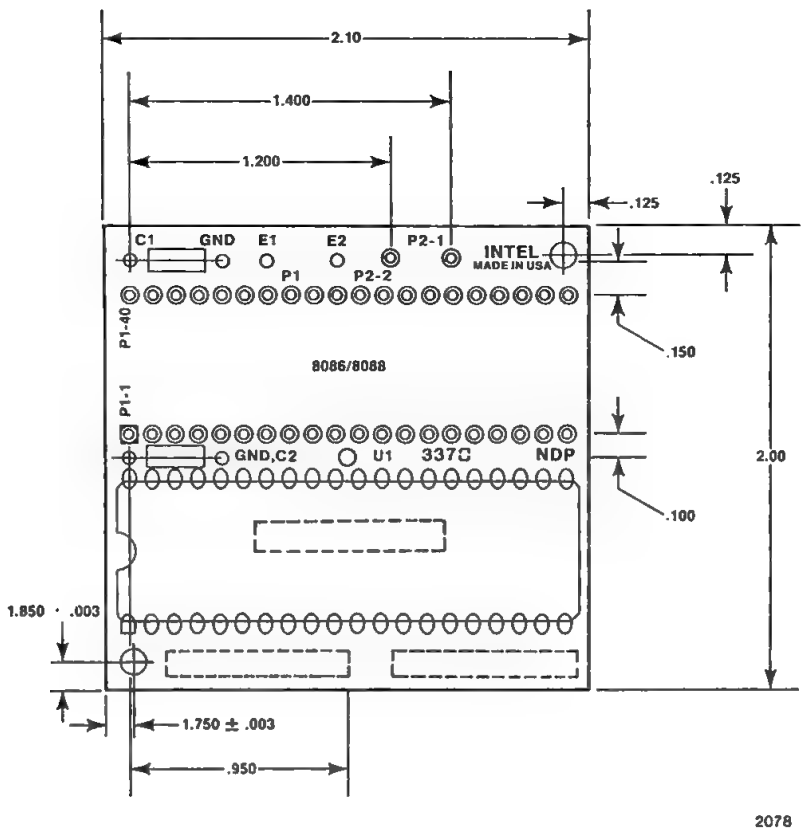


Figure 2-1. Board Dimensions (Inches)

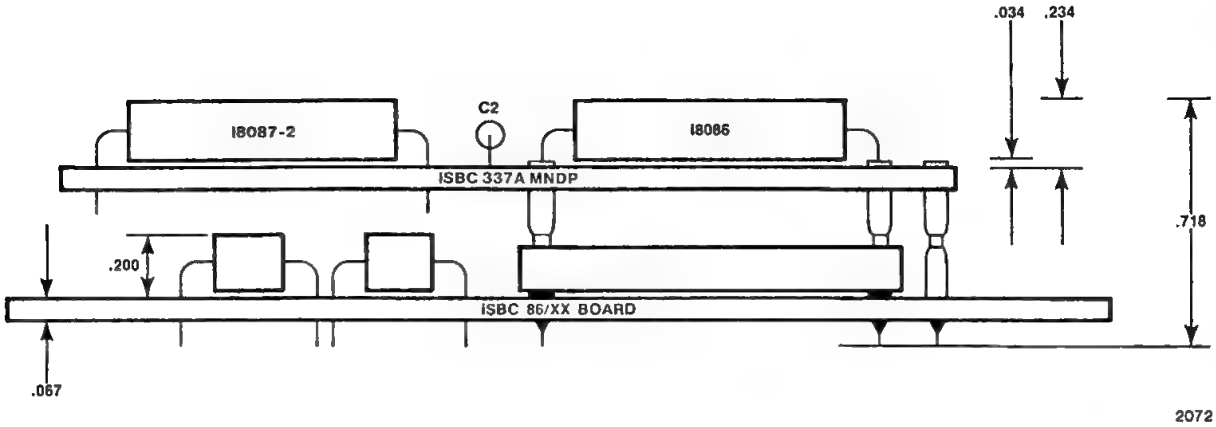


Figure 2-2. Mounting Clearances (Inches)

Table 2-1. Connectors P1 and P2 Pin Assignments

Connector P1			Connector P1		
PIN	Mnemonic	Description	PIN	Mnemonic	Description
1	Vss	Signal Ground	21	RESET	Reset
2	ADE	I/O Address and Data E	22	READY	Data Ready For Transfer
3	ADD	I/O Address and Data D	23	TEST/BUSY	Busy status from 8087-2
4	ADC	I/O Address and Data C	24	QS1	Instruction Queue Status 1 Output
5	ADB	I/O Address and Data B	25	QS0	Instruction Queue Status 0 Output
6	ADA	I/O Address and Data A	26	S0	Status 0 Output
7	AD9	I/O Address and Data 9	27	S1	Status 1 Output
8	AD8	I/O Address and Data 8	28	S2	Status 2 Output
9	AD7	I/O Address and Data 7	29	LOCK	Lock Out System Bus
10	AD6	I/O Address and Data 6	30	RQ/GT1	Request Grant Signal 1
11	AD5	I/O Address and Data 5	31	RQ/GT0	Request Grant Signal 0
12	AD4	I/O Address and Data 4	32	RD	Read Cycle
13	AD3	I/O Address and Data 3	33	MN/MX	Minimum or Maximum mode
14	AD2	I/O Address and Data 2	34	BHE/S7	Byte High Enable/Status Bit 7
15	AD1	I/O Address and Data 1	35	A13/S6	Address Line 13/Status Bit 6
16	AD0	I/O Address and Data 0	36	A12/S5	Address Line 12/Status Bit 5
17	NMI	Non-Maskable Interrupt	37	S11/S4	Address Line 11/Status Bit 4
18	INTR	Interrupt Request	38	S10/S3	Address Line 10/Status Bit 3
19	CLK	Clock Input	39	ADF	I/O Address and Data F
20	Vss	Signal Ground	40	Vcc	+5 Volts
Connector P2			**Connector P2**		
PIN	Mnemonic	Description	PIN	Mnemonic	Description
1	MINT	Interrupt Output	2	RQ/GT1A	Request Grant to 8087-2

x-903

Table 2-2. Connectors P1 and P2 Signal Functions

Signal	Description
ADØ - ADF (Input/ Output 3-State)	Address/Data. These lines constitute the time multiplexed address and data bus. ADØ is analogous to BHE/ for the lower byte of the data bus (ADØ-AD7). ADØ is low during T1 when a byte is to be transferred on the lower portion of the bus during memory or I/O operations. Eight-bit oriented devices tied to the lower half would normally use ADØ to continue chip select functions. These lines are active high and float to tri-state during interrupt acknowledge and local bus hold acknowledge.
A1Ø/S3-A13/S6 (Output 3-State)	Address/Status. During T1 these are the four most significant address lines for memory operations. During memory and I/O operations, status information is available on these lines during T2, T3, TW, and T4. These lines float to tri-state during interrupt acknowledge and local bus hold acknowledge. The status of the interrupt enable flag bit (S5) is updated at the beginning of each CLK cycle. A 3/S6 is monitored by the 8Ø87-2 to indicate when the 8Ø86/88 has control of the local bus. A11/S4 and A1Ø/S3 indicate which relocation register on the CPU is presently being used for data accessing.
BHE/S7 (Output 3-State)	Byte High Enable/Status Bit 7. During T1 the byte high enable (BHE/) signal is used to enable data for the most significant half of the data bus (AD8-ADF). Eight-bit oriented devices tied to the upper half of the bus would normally use BHE/ to condition chip select functions. BHE/ is low during T1 for read, write, and interrupt acknowledge cycles when a byte is to be transferred on the high portion of the bus. The S7 status information is available during T2, T3, and T4. BHE/ is active low and floats to a tri-state condition when the CPU is in hold. BHE/ is low during T1 for the first interrupt acknowledge cycle.
CLK (Input)	Clock. The clock provides the basic timing for the processor and bus controller. It is asymmetrical with a 33% duty cycle.
INTR (Input)	Interrupt Request. This pin is a level triggered input which is sampled during the last clock cycle of each instruction to determine if the processor should enter into an interrupt acknowledge operation. It is used only by the CPU.

(continued)

Table 2-2. Connectors P1 and P2 Signal Functions (continued)

Signal	Description
LOCK/ (Output 3-State)	LOCK. This output signal indicates that other system bus masters are not to gain control of the system bus while the signal is active. It is used only by the CPU.
MINT (P2) (Output)	Math Interrupt. This signal is the interrupt output of the 8Ø87-2. It represents mathematical errors or exceptions within the iSBC 337A board. It is normally connected to the interrupt control circuits of the base board.
MN/$\overline{MX}$ (Input)	Minimum/Maximum Mode. This pin determines whether the CPU is to operate in minimum or maximum mode. It is wired for maximum mode on the iSBC 337A MNDP.
NMI (Input)	Non-maskable Interrupt. This pin is an edge triggerred input which causes a type 2 interrupt. It is used only by the CPU.
QSØ-QS1 (Output)	Queue Status. QS1 and QSØ provide status to allow external tracking of the internal 8Ø86 instruction queue. QSØ and QS1 are encoded as follows: QS1 QSØ Ø Ø No Operation Ø 1 First Byte of OP Code from Queue 1 Ø Empty the Queue 1 1 Subsequent Byte from Queue The queue status is valid during the clock cycle after which the queue operation is performed. These pins used by the 8Ø87 are not available for use elsewhere in the system.
RD/ (Output 3-State)	READ. This signal indicates that the processor is performing a memory read or I/O read cycle. It is not used by the iSBC 337A MNDP.
READY (Input)	Ready for Data Transfer. This signal is the acknowledgement from the addressed memory or I/O device that is ready for the data transfer. The RDY signal from memory I/O is synchronized by the 8284A Clock Generator to form READY.

(continued)

Table 2-2. Connectors P1 and P2 Signal Functions (continued)

Signal	Description
RESET (Input)	RESET. The RESET input causes the 8Ø87-2 and the processor to immediately terminate their present activities. The signal must be active high for at least four clock cycles. The CPU and the 8Ø87-2 restart execution when the reset signal returns low. The reset signal is internally synchronized in the CPU.
$\overline{RQ}/\overline{GT}Ø$ (Input/Output)	Request/Grant. This pin has the same function as RQ/GT1 but has a higher priority. It is used only by the CPU.
$\overline{RQ}/\overline{GT1}$ (Input/Output)	Request/Grant. This pin is connected to the 8Ø87-2's RQ/GTØ pin on the iSBC 337A board. The request/grant signal is used by the CPU and the 8Ø87-2 MNDP to exchange control of the local bus. This pin is not available for use elsewhere in the system.
$\overline{RQ}/\overline{GT1A}$ (P2) (Input/Output)	Request/Grant One A. This pin has the same function as RQ/GTØ but has a lower priority. RQ/GT1A is connected to RQ/GT1 of the 8Ø87-2. The 8Ø87-2 passes the request and grant signals on to the 8Ø86 via the 8Ø87-2's RQ/GTØ pin (connected to the RQ/GT1 pin of the 8Ø86). This allows other co-processors to acquire the local bus.
SØ-S2 (Output 3-State)	Status Lines. These three status lines furnish information to the 8288 Bus Controller and the 8289 Bus Arbiter. The status information is valid during T4, T1, and T2 and is returned to the passive state, during TW, when the ready signal is high. This status is used by the 8288 Bus Controller to generate all memory and I/O access control signals. A change in S2, S1, or SØ during T4 indicates the beginning of a bus cycle, and a return to the passive state in T3 or TW indicates the end of the bus cycle. This pin is used by the 8Ø87-2 and is not available for use elsewhere in the system.
TEST/BUSY (Input)	Test. This pin is connected to the 8Ø87-2's BUSY pin on the iSBC 337A board. When this signal is high it causes the CPU to wait in an idle state when the CPU executes a wait instruction. This pin is used by the 8Ø87-2 and is not available for use elsewhere in the system.
V_{CC} and V_{SS}	Voltage Supply and Ground. +5 volts dc and signal ground.

***

CHAPTER 3
PROGRAMMING INFORMATION

3.1 INTRODUCTION

This chapter describes the programming for the iSBC 337A MULTIMODULE NDP board. Included are sections on the instruction set, data formats, status format, and programming examples.

3.2 PROGRAMMING INTERFACE

The combination of an iSBC 86 or iSBC 88 single board computer and an iSBC 337A MNDP generally appears to the programmer as a single machine. The iSBC 337A MNDP adds new data types, registers, and instructions to the host iSBC. The programming languages and the co-processor architecture take care of most inter-processor coordination automatically.

Computations in the 8Ø87-2 are all done in the temporary real format (see section 3.4.1). The numbers are translated to this format internally when loaded and are translated back to the original format when stored in memory. The computations in the 8Ø87-2 are centered around the co-processor's register stack. These 8Ø-bit registers provide the equivalent capacity of 4Ø of the 16-bit registers found in typical CPU's. The 8Ø87-2 register set can be accessed both as a stack, with instructions operating implicitly on the top one or two stack elements, and as a fixed register set, with instructions operating on explicitly designated registers.

Two features of the 8Ø87-2 hardware further simplify numeric application programming. First, the 8Ø87-2 is invoked directly by the programmer's instructions. There is no need to write instructions that address the 8Ø87-2 as an I/O device, or to incur the overhead of setting up a DMA operation to perform data transfers. Second, the 8Ø87-2 automatically detects exception conditions that can potentially cause a miscalculation at run-time. The 8Ø87-2 exception handlers are automatically invoked by default to field these exceptions so that a reasonable result is produced and execution may proceed without program intervention. Alternately, the 8Ø87-2 can interrupt the host CPU and thus trap to a user procedure when an exception is detected.

3.3 INSTRUCTION SET

The 8Ø87-2 instructions are intermixed with the host CPU instructions in a single instruction stream that is fetched by the host CPU. By monitoring the status signals emitted by the host CPU, the 8Ø87-2 control unit can determine when an instruction is being fetched. When the instruction byte or word becomes available on the local bus, the 8Ø87-2 control unit monitors the bus in parallel with the host CPU and obtains that portion of the instruction. Table 3-1 is an abbreviated description of the instructions that are performed by the 8Ø87-2 on the iSBC 337A MULTIMODULE NDP board. A more detailed description of each instruction may be found in the 8Ø86/8Ø87 Macro Assembly Language Reference Manual.

Table 3-1. 8Ø87-2 NDP Instructions

Mnemonic	Instruction Description
	Data Transfer
FLD FILD FBLD	These instructions perform an 8Ø87-2 load (push) of the source to the top of the floating-point stack.
FST FIST FSTP FISTP FBSTP	These instructions perform an 8Ø87-2 store of the top of the floating-point stack in the destination specified.
FXCH	This instruction exchanges the floating-point stack top with the floating-point stack element.
	Basic Arithmetic and Operands
FABS	This instruction changes the top element of the 8Ø87-2 floating-point stack to its absolute value by making its sign a positive value.
FADD FIADD FADDP	These instructions perform an 8Ø87-2 floating-point (or integer) addition of the source to the destination, leaving the result in the destination.
FCHS	This instruction performs a sign change of the 8Ø87-2 floating-point stack top.

(continued)

Table 3-1. 8Ø87-2 NDP Instructions (continued)

Mnemonic	Instruction Description
	Basic Arithmetic and Operands (continued)
FDIV FIDIV FDIVP	These instructions perform an 8Ø87-2 floating-point (or integer) division of the destination by the source, leaving the result in the destination.
FDIVR FIDIVR FDIVRP	These instructions perform a reverse 8Ø87-2 floating-point (or integer) division, where the source is divided by the destination, and the result is left in the destination.
FMUL FIMUL FMULP	These instructions perform an 8Ø87-2 floating-point (or integer) multiplication of the destination by the source, leaving the result in the destination.
FPREM	This instruction calculates the remainder of the top two 8Ø87-2 floating-point stack elements (i.e., stack top (ST(Ø)) modular next-to-top (ST(1)) stack element), returning the result to the stack top. The next-to-top stack element is left unchanged.
FRNDINT	This instruction rounds the top stack element of the 8Ø87-2 floating-point stack and returns it to the stack top.
FSCALE	This instruction adds the next-to-top element of the 8Ø87-2 floating-point stack to the exponent of the stack top, leaving the result in the stack top. The next-to-top element is left unchanged.
FSQRT	This instruction returns the square root of the 8Ø87-2 floating-point stack top to the stack top.
FSUB FISUB FSUBP	These instructions perform an 8Ø87-2 floating-point (or integer) subtraction of the source from the destination leaving the result in the destination.
FSUBR FISUBR FSUBRP	These instructions perform a reverse 8Ø87-2 floating-point (or integer) subtraction, where the destination is subtracted from the source, and the result is left in the destination.

(continued)

Table 3-1. 8Ø87-2 NDP Instructions (continued)

Mnemonic	Instruction Description
	Basic Arithmetic and Operands (continued)
FXTRACT	This instruction extracts the exponent and significand from the 8Ø87-2 floating-point stack top, returning the exponent (in floating-point format) to the stack top, then pushing the significand (scaled between 1.Ø and 2.Ø) onto the stack.
	Comparison
FCOM FICOM FCOMP FICOMP FCOMPP	These instructions perform an 8Ø87-2 floating-point (or integer) compare of the source operand and the top of the floating-point stack. The compare affects the 8Ø87-2 status flags and stack pointer.
FTST	This instruction compares the top element of the 8Ø87-2 floating-point stack to a floating-point zero (+Ø.Ø).
FXAM	This instruction examines the top element of the 8Ø87-2 floating-point stack and sets the status bits.
	Transcendental
FPATAN	This instruction calculates the arctangent of Y/Z, where Z is the 8Ø87-2 floating-point stack top (ST(Ø)) and Y is the next-to-top stack element (ST(1)). The stack is popped and the result returned on the (new) stack top.
FPTAN	This instruction computes the function Y/X = tan Z, where Z is the 8Ø87-2 floating-point stack top. Y is returned to the stack top, then Z is pushed onto the stack.
FYL2X	This instruction multiplies Y, the next-to-top of stack element (ST(1)), times log base 2 of X, the stack top (ST(Ø)). The stack is popped and the result returned on the (new) stack top.

(continued)

Table 3-1. 8Ø87-2 NDP Instructions (continued)

Mnemonic	Instruction Description
	Transcendental (continued)
FYL2XP1	This instruction multiplies Y, the next-to-top of stack element (ST(1)), times log base 2 of X, the (stack top), plus one. The stack is popped and the result returned to the (new) stack top.
F2XM1	This instruction raises two to the power of X, (stack top), minus one (i.e., $Y = 2^X - 1$).
	Constants
FLDLG2	This instruction performs an 8Ø87-2 load (push) of the real constant log base 1Ø of two ($\log_{1Ø}$ (2.Ø)) to the top of the floating-point stack.
FLDLN2	This instruction performs an 8Ø87-2 load (push) of the real constant log base e of two (ln 2.Ø) to the top of the floating-point stack.
FLDL2E	This instruction performs an 8Ø87-2 load (push) of the real constant base log base 2 of e ($\log_2$ (e)) to the top of the floating-point stack.
FLDL2T	This instruction performs an 8Ø87-2 load (push) of the real constant log base 2 of ten ($\log_2$ (1Ø.Ø)) to the top of the floating-point stack.
FLDPI	This instruction performs an 8Ø87-2 load (push) of the real constant pi to the top of the floating-point stack.
FLDZ	This instruction performs an 8Ø87-2 load (push) of the real constant zero (+Ø.Ø) to the top of the floating-point stack.
FLD1	This instruction performs an 8Ø87-2 load (push) of the real constant one (+1.Ø) to the top of the floating-point stack.

(continued)

Table 3-1. 8Ø87-2 NDP Instructions (continued)

Mnemonic	Instruction Description
	Processor Control
FCLEX FNCLEX	These instructions clear any 8Ø87-2 exception flag bits that are set in the status word.
FDISI FNDISI	These instructions set the 8Ø87-2 interrupt enable mask, disabling all 8Ø87-2 interrupts.
FDECSTP	This instruction decrements the 8Ø87-2 stack pointer.
FFREE	This instruction sets the tag bits, for the register specified in the operand, to one (register empty). This is a special case of the load instruction.
FINCSTP	This instruction increments the 8Ø87-2 stack pointer by one.
FINIT FNINIT	These instructions initialize the 8Ø87-2.
FLDCW	This instruction loads the 8Ø87-2 control word (2 bytes) from the memory source specified into the status word.
FLDENV	This instruction loads the 8Ø87-2 status and recovery information (14 bytes) from the memory source specified.
FNOP	This instruction performs an 8Ø87-2 NOP instruction. It is identical to storing the floating-point stack top into itself (i.e., FST ST(Ø)).
FRSTOR	This instruction restores the 8Ø87-2 context (status information, recovery information, and stack, 94 bytes) from the memory source specified.
FSAVE FNSAVE	These instructions save the 8Ø87-2 context (status information, recovery information, and stack, 94 bytes) into the memory location specified.
FENI FNENI	These instructions clear the 8Ø87-2 interrupt enable mask in the control word, thus enabling interrupts.

(continued)

Table 3-1. 8Ø87-2 NDP Instructions (continued)

Mnemonic	Instruction Description
	Processor Control (continued)
FSTCW FNSTCW	These instructions store the 8Ø87-2 control word (2 bytes) into the memory destination specified.
FSTENV FNSTENV	These instructions store the 8Ø87-2 status and recovery information (14 bytes) into the memory destination specified.
FSTSW FNSTSW	These instructions store the 8Ø87-2 status word (2 bytes) into the memory destination specified.
FWAIT	This instruction performs an 8Ø86 WAIT for the purpose of synchronization with the 8Ø87-2.

3.4 DATA FORMATS

The iSBC 337A MNDP supports three data sets: real, integer, and BCD.

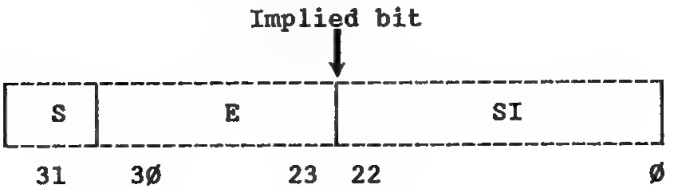
3.4.1 REAL

Real numbers can be presented to the 8Ø87-2 in three formats: single precision, double precision, and file precision (temporary real).

NOTE

The 8Ø87-2 stores all numbers internally as file precision numbers. They are converted back to their original format (or another specified format) when stored externally in memory.

Single Precision Data Format

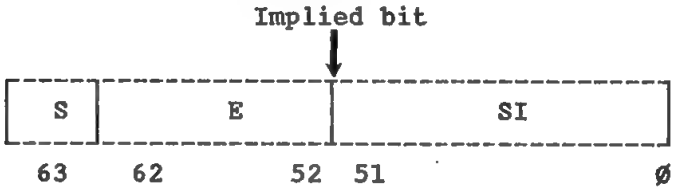


Bit 31:
S = Sign of the significand, 1 represents negative and Ø represents positive.

Bits 23-3Ø:
E = These 8 bits represent a biased exponent. The bias is $2^7 - 1$ (127). The range of the exponent is -126 to +127.

Bits Ø-22:
SI = 23-bit significand. Together with the sign bit, the significand represents a signed fraction in sign-magnitude notation. There is an implied binary point to the left of the most significant bit (bit 22) of the significand. In other words, the significand is assumed to be a 24-bit normalized quantity and the most significant bit, which will always be 1 due to normalization, is implied. The binary point is between the implied bit and bit 22 of the significand.

Double Precision Data Format



Bit 63:
S = Sign of the significand, 1 represents negative and Ø represents positive.

Bits 52-62:
E = These 11 bits represent a biased exponent. The bias is $2^{1Ø} - 1$ (1Ø23). The range of the exponent is -1Ø22 to +1Ø23.

Bits Ø-51:
SI = 52-bit significand. Together with the sign bit, the significand represents a signed fraction in sign-magnitude notation. There is an implied binary point to the left of the most significant bit (bit 51) of the significand. In other words, the significand is assumed to be a 53-bit normalized quantity and the most significant bit, which will always be 1 due to normalization, is implied. The binary point is between the implied bit and bit 51 of the significand.

File Precision

Bit 79:
S = Sign of the significand, 1 represents negative and Ø represents positive.

Bits 64-78:
E = These 15 bits represent a biased exponent. The bias is $2^{14} - 1$ (16383). The range of the exponent is -16382 to +16383.

Bits Ø-63:
SI = 64-bit significand. Together with the sign bit, the significand represents a signed fraction in sign-magnitude notation. There is an implied binary point to the right of the most significant bit (bit 63) of the significand. In other words, the significand is assumed to be a 64-bit quantity. If bit 63 is 1, the number has been normalized. A file precision number inside the 8Ø87-2 which has not been normalized is called un-normalized. A file precision number in memory which is not normalized is called a de-normalized number.

NOTE

In file precision only, the implied binary point is located between bits 62 and 63 (the two most significant bits of the fraction) with a one always located at bit 63 if the number is normalized.

The following is an example of how the single precision floating-point number 127.133H would be represented.

The bit pattern is:

1	2	7 . 1	3	3	H
ØØØ1	ØØ1Ø	Ø111.ØØØ1	ØØ11	ØØ11	B

The binary point is now moved to the right of the most significant bit whose value is equal to "1".

The new bit pattern is:
ØØØ1.ØØ1Ø Ø111 ØØØ1 ØØ11 ØØ11

The binary point was moved eight places:
$1.ØØ1ØØ111ØØØ1ØØ11ØØ11 \times 2^{8}$ or $1.27133 \times 2^{8}H$

To organize this number to load into the 8Ø87-2 NDP, the bit pattern is:

S	EXPONENT	SIGNIFICAND (23 bits)
Ø	1ØØØØ111	ØØ1ØØ111ØØØ1ØØ11ØØ11ØØØ

The exponent is derived by adding the eight places the binary point was moved to the bias of 127.

Example:

```
    Ø111     1111
  + ØØØØ     1ØØØ
    1ØØØ     Ø111
```

3.4.2 INTEGER

There are three forms of integer numbers that the iSBC 337A board can handle. All three forms use twos complement arithmetic:

WORD -- 16 bits
SHORT -- 32 bits
LONG -- 64 bits

Word Format

Bit 15:
S = Sign of the integer, 1 represents negative and Ø represents positive.

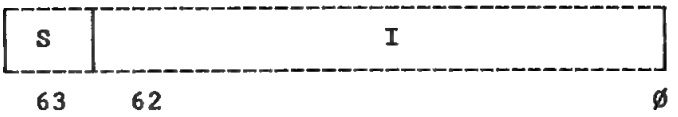
Bits Ø-14:
I = These 15 bits represent an integer value from Ø to 7FFF hexadecimal, in twos complement notation.

Short Format

Bit 31:
S = Sign of the integer, 1 represents negative and Ø represents positive.

Bits Ø-3Ø:
I = These 31 bits represent an integer value from Ø to 7FFFFFFF hexadecimal, in twos complement notation.

Long Format



Bit 63:
S = Sign of the integer, 1 represents negative and Ø represents positive.

Bits Ø-62:
I = These 63 bits represent an integer value from Ø to 7FFFFFFFFFFFFFFF hexadecimal, in twos complement notation.

3.4.3 PACKED BINARY CODED DECIMAL

The iSBC 337A board directly manipulates Packed Binary Coded Decimal data, which is useful for applications that require BCD numbers (e.g., business applications). The 8Ø87-2 accepts an 8Ø bit number representing a signed 18 digit packed BCD number:

Bit 79:

S = Sign of the packed BCD number, 1 represents negative and Ø represents positive.

Bits 72-78:
Must be zero.

Bits Ø-71:
I = These 72 bits represent a packed BCD value (4 bits per decimal digit) from Ø to $\pm(1x1Ø^{19})-1$.

3.5 STATUS FORMAT

The environment of the iSBC 337A MNDP is stored in memory as 7 contiguous words. The first 3 words contain the status information and the last 4 contain the recovery information. Figure 3-1 illustrates the form that the status information takes when stored in memory.

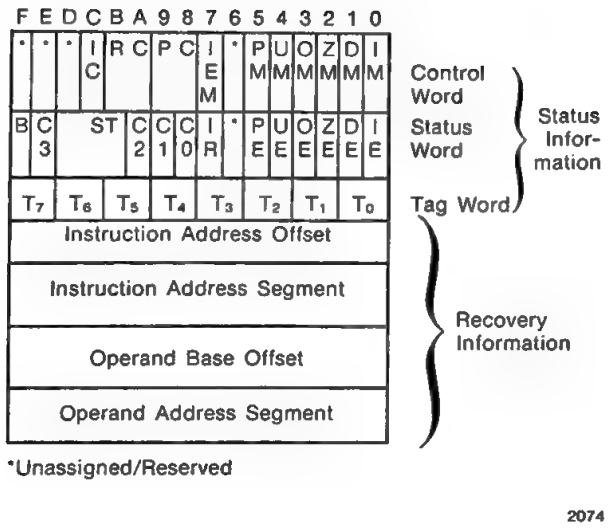


Figure 3-1. Status Information Format

3.5.1 CONTROL WORD

The control word specifies the mode of operation for the 8Ø87-2, contains some special controls, and also contains the interrupt mask bits. The control word bit definition is:

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	Ø
*	*	*	IC	RC		PC		IEM	*	PM	UM	OM	ZM	DM	IM

Bits D-F:
These bits are reserved.

Bit C:
IC = Infinity Control. The 8Ø87-2 employs two methods of closing the number system, either projective or affine closure. In the projective closure, infinity is unsigned. In affine closure, infinity carries a + or - sign. The infinity control bit definition is:

Ø = projective closure, 1 = affine closure (Default is Ø).

Bits A-B:
RC = The contents of the rounding control field determines the rounding algorithm to be used in producing a result. The most common setting is round to even. Rounding down (toward negative infinity) or up (toward positive infinity) should be used when the result must be precisely controlled (internal arithmetic, elementary functions, etc.).

B	A		
Ø	Ø	Round to nearest even	(default)
Ø	1	Round down	
1	Ø	Round up	
1	1	Chop (truncate)	

Bits 8-9:
PC = The contents of the precision control field determines the number of bits of precision of the significand to retain in the arithmetic results. Each result is rounded to the appropriate precision. The precision of the results is thus independent of the precision of the operand.

9	8		
Ø	Ø	24 bits	
Ø	1	(Reserved)	
1	Ø	53 bits	
1	1	64 bits	(default)

Bit 7:
IEM = The interrupt enable mask bit is the common mask bit that masks the IR bit in the status word from sending an interrupt to the processor. When set to zero, it allows the IR bit to indicate an exception to the processor if any unmasked exception bit is set. The master interrupt bit is set by the disable interrupt instruction and reset by the enable interrupt. The load control instruction can also set and reset these bits. The interrupt enable mask bit definition is:

Ø = unmask, 1 = mask (Default is 1).

Bit 6:
This bit is reserved.

Bits Ø through 5:
These bits mask the individual exception flag bits. The names and indication of each exception flag bit is:

Bit 5:
PM = Precision Exception Flag Bit: Ø = unmasks, 1 = masks (Default is 1).

Bit 4:
UM = Underflow Exception Flag Bit: Ø = unmasks, 1 = masks (Default is 1).

Bit 3:
OM = Overflow Exception Flag Bit: Ø = unmasks, 1 = masks (Default is 1).

Bit 2:
ZM = Zero Divisor Exception Flag Bit: Ø = unmasks, 1 = masks (Default is 1).

Bit 1:
DM = De-normalized Operand Exception Flag Bit: Ø = unmasks, 1 = masks (Default is 1).

Bit Ø:
IM = Invalid Operation Exception Flag Bit: Ø = unmasks, 1 = masks (Default is 1).

3.5.2 STATUS WORD

The status word holds the current status of the 8Ø87-2. Some of the upper bits (8, 9, A, E, and F) will be used for conditional branching after operands have compared, and for polling the status of the 8Ø87-2. The lower 8 bits (Ø - 7) hold the exception information for the instruction being performed and will be used for exception handling information.

The status word is read into memory with the floating-point store status word (FSTSW) instruction. The floating-point clear exceptions (FCLEX) instruction clears the exception flag bits. The definition for the bits in the status word is:

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	Ø
B	C3	ST			C2	C1	CØ	IR	*	PE	UE	OE	ZE	DE	IE

Bit F:

B = The busy (B) and interrupt (IR) flag bits denote the execution status of the 8Ø87-2. When the 8Ø87-2 begins executing an instruction, the B bit sets. Upon successful completion of the instruction (if any exception flags were encountered and they were masked), the B bit is reset. However, if an unmasked exception occurred, the IR bit is set and the B bit remains set until the exception is reported through the interrupt system. Since the B bit has the same setting as the pin that is tested by the wait instruction that is inserted by the assembler, no iSBC 337A MNDP instruction can gain control of the 8Ø87-2 until a FCLEX instruction (which clears the flags) is executed by an exception handler. The busy flag bit definition is:

Ø = not busy, 1 = busy.

Bits E, A, 9, 8:

C3 to CØ = Several 8Ø86 instructions post their results to the condition code bits. The principal use of the condition codes is for conditional branching. Table 3-2 lists the meaning of the various settings of the condition codes.

Table 3-2. Condition Code Setting

Condition Code				Interpretation
C3	C2	C1	C0	
0	0	0	0	+ Unnormal
0	0	0	1	+ NAN
0	0	1	0	- Unnormal
0	0	1	1	- NAN
0	1	0	0	+ Normal
0	1	0	1	+ Infinity
0	1	1	0	- Normal
0	1	1	1	- Infinity
1	0	0	0	+ 0
1	0	0	1	Empty
1	0	1	0	- 0
1	0	1	1	Empty
1	1	0	0	+ Denormal
1	1	0	1	Empty
1	1	1	0	- Denormal
1	1	1	1	Empty

x-904

Bits B-D:

ST = The top stack pointer contains the address of the register that is the current top of the stack. During the power-on sequence this register (top of stack) is initialized to zero. Its manipulation beyond that point is user controlled.

Bit 7:
IR = The interrupt request flag bit is set at the end of execution of any instruction that caused an unmasked error bit to be set. If the IEM bit (interrupt-enable mask bit) is clear, bit 7 being set will cause an interrupt signal to be sent out on the INT (interrupt) line of the iSBC 337A MNDP to interrupt the CPU at the end of the instruction being executed. The interrupt request flag bit definition is:

Ø = no error, 1 = error.

Bit 6:
Reserved bit.

Bits 5 through Ø:
Exception Flags. Each exception flag bit is set upon the detection of a specific variety of exception condition. The exception flags are cleared by the execution of the clear exception (FCLEX), load environment (FLDENV), restore all (FRSTOR), initialize (FINIT), and save all (FSAVE) instructions. If an exception flag is set to 1, it indicates that this exception has occurred. All the exception flags are set to zero on power up. The names and explanation of each exception flag bit are as follows:

Bit 5:
PE = Precision. This exception flag indicates that the result of an operation is not exactly representable in the destination format. The 8Ø87-2 will round the number and set this flag.

Bit 4:
UE = Underflow. This exception flag indicates that the result of an operation is too small to be represented in the destination format.

Bit 3:
OE = Overflow. This exception flag indicates that the result of an operation is too large to be represented in the destination format.

Bit 2:
ZE = Zero divide. This exception flag indicates that an attempt was made to divide zero into some number other than zero. (Dividing zero by zero results in the invalid operation exception flag being set.)

Bit 1:
DE = De-normalized Operand. This exception flag indicates that a de-normalized number has been fetched.

Bit Ø:
IE = Invalid Operation. This exception flag indicates that an attempt was made to perform an invalid operation.

3.5.3 TAG WORD

The tag word contains eight tags (two bits each). Each tag word describes the contents of the internal register T_x. The principal function of the tag word is to optimize the 8Ø87-2 performance under certain conditions and programmers ordinarily will not be concerned with it.

The meaning of the bits in the tags are:

ØØ = register contains a valid, non-zero quantity.
Ø1 = register contains a zero.
1Ø = register contains an invalid, de-normalized, or infinite number.
11 = register is empty (default).

3.6 EXAMPLE PROBLEMS FOR FLOATING POINT

Two sample programs are illustrated. The first example calculates the hypotenuse of a right triangle, and the second example solves a quadratic equation.

Example 1:

```
:F1:WAYNE.ASM  PAGE 1

NAME WAYNE_SCHMIDT_SAMPLE_PROGRAM

CGROUP  GROUP   CODE
DGROUP  GROUP   DATA

ASSUME  CS:CGROUP,DS:DGROUP

;THIS IS A SAMPLE PROGRAM TO DEMONSTRATE THE USE OF THE 8Ø87 NUMERIC DATA
;PROCESSOR.  IT WILL CALCULATE THE HYPOTENUSE OF A RIGHT TRIANGLE.
;LET THE SIDES =  a (opposite); b (adjacent); c (hypotenuse).
;THE PROGRAM WILL SOLVE FOR : a^2 + b^2 = c^2.
;a, b, and c ARE DATA TERMS.
;a = 3, b = 4, c = unknown.

a               EQU     3.Ø
b               EQU     4.Ø

DATA            SEGMENT WORD PUBLIC 'DATA'    ;DECLARE THE DATA SEGMENT
SIDEa           DT      a             ;DEFINE A FLOATING POINT NUMBER THREE
                                      ;IN TEMPORARY REAL FORM
SIDEb           DT      b             ;DEFINE A FLOATING POINT NUMBER FOUR
                                      ;IN TEMPORARY REAL FORM
HYPOT           DB      1Ø DUP(?)     ;DEFINE A STORAGE AREA FOR THE RESULT
CONTROL_WORD    DW      1 DUP(?)      ;DEFINE A STORAGE AREA FOR THE CONTROL
                                      ;WORD
```

```
:F1:WAYNE.ASM  PAGE 2

STATUS_WORD     DW      1 DUP(?)        ;DEFINE A STORAGE AREA FOR THE
STATUS
                                        ;WORD
DATA            ENDS

CODE            SEGMENT WORD PUBLIC 'CODE'

PYTHAGOREAN_THEOREM   PROC          NEAR
                FINIT                   ;INITIALIZE THE 8Ø87
                MOV     CONTROL_WORD,3FFFH
                                        ;SET THE CONTROL WORD FOR FILE
                                        ;PRECISION,
                                        ;ROUND TO EVEN, AND MASK OFF THE
                                        ;INTERRUPTS
                FLDCW   CONTROL_WORD    ;LOAD THE CONTROL WORD
                FLD     TBYTE PTR SIDEa ;LOAD SIDE a
                FMUL    ST,ST(Ø)        ;SQUARE SIDE a
                FLD     TBYTE PTR SIDEb ;LOAD SIDE b
                FMUL    ST,ST(Ø)        ;SQUARE SIDE b
                FADD                    ;ADD SIDES a AND b AND POP THE
                                        ;STACK
                                        ;THE RESULT IS NOW ON THE TOP OF
                                        ;THE STACK
                FSQRT                   ;FIND THE SQUARE ROOT OF THE RESULT
                                        ;OF a^2 + b^2
                FSTSW   STATUS_WORD     ;GET A COPY OF THE STATUS WORD
                MOV     AX,STATUS_WORD  ;COPY THE STATUS TO THE AX REGISTER
                AND     AL,ØBFH         ;MASKS OFF BIT 6
                OR      AL,AL           ;CHECK FOR ERRORS OR EXCEPTIONS
                JNZ     ERRORS          ;IF ERRORS EXIST, THEN HANDLE IT
                                        ;IF NOT, THEN GET THE RESULT
                FSTP    TBYTE PTR HYPOT ;RETRIEVE THE VALUE FROM THE 8Ø87
                                        ;AND PLACE IN MEMORY
PYTHAGOREAN_THEOREM   ENDP

ERRORS          PROC    NEAR            ;USER DEFINED ERROR HANDLER
                HLT
ERRORS          ENDP

CODE            ENDS
                END
```

Example 2:

For the second example, the following equation will be computed.

$$ax^2 + bx + c = Ø$$

Example 2 (continued)

Rearranging the equation to solve for x:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The equation can be solved in 5 major steps where:

1) $N = 2(a)$

2) $M = 4(a)(c)$

3) $L = b^2$

4) $K = \sqrt{L-M}$

5) $x = \frac{-b \pm K}{N}$

The values of variables "a", "b", and "c":

(a=1, b=3, c=2)

The eight register stack files of the 8Ø87-2 make it easy to do more than one calculation at a time without a memory reference.

Figure 3-2 is an overall flow diagram for the steps to calculate the equation. Figure 3-3 is a detailed flow diagram of the steps to calculate the equation. Table 3-3 details the register stack after each operation. The step numbers running horizontally correspond to the step numbers in the flow chart of Figure 3-3.

3.6.1 PROBLEM TWO PROGRAM DETAIL

The step numbers in the following detailed program correspond with the step numbers in figure 3-3. The following is a detailed program of the routine to calculate the roots of the Quadratic Equation listed previously. The numbers depicted (a=1, b=3, c=2) will be used here. The program will use real numbers in a single precision form.

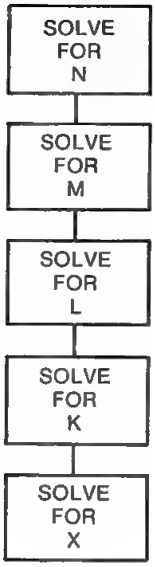


Figure 3-2. Overall Programming Example Flow Chart

Table 3-3. Stack Image Per Stack

Stack Register	Step Sequence														
	1	2	3	4	5	6A	6B	7A	7B	8	9A	9B	1ø	11A	11B
ST(ø)	2	N	b	L	4	4A	M	L-M	K	K	-b+K	$\frac{-b+K}{N}$	-b	-b-K	$\frac{-b-K}{N}$
ST(1)	a	a	a	a	a	c	L	-b	-b	-b	-b	-b	N	N	K
ST(2)	c	c	c	c	c	L	-b	N	N	N	N	N	K	K	
ST(3)	4	4	4	4	L	-b	N			K	K	K			
ST(4)	-b	-b	-b	-b	-b	N									
ST(5)	b	b	N	N	N										
ST(6)															
ST(7)															

x-902

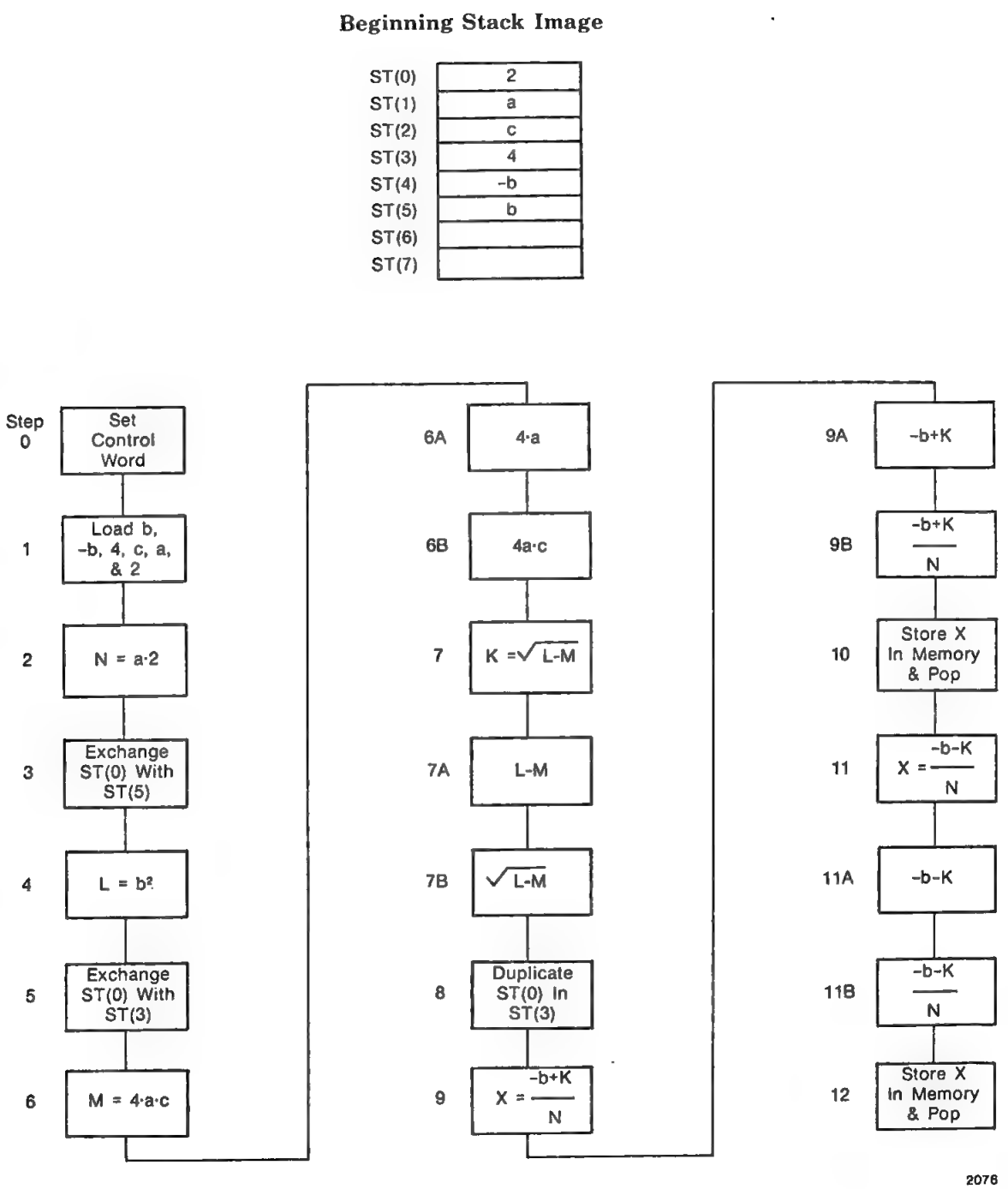


Figure 3-3. Detailed Programming Example Flow Chart

```
:F1:SAMPLE.ASM  PAGE 1

NAME SAMPLE_PROG_2

CGROUP  GROUP   CODE
DGROUP  GROUP   DATA

ASSUME  DS:DGROUP,CS:CGROUP

;THIS PROGRAM IS AN EXAMPLE OF HOW TO USE FLOATING POINT INSTRUCTIONS.
;WHILE A MORE EFFICIENT PROGRAM MAY BE WRITTEN, THIS ONE CAN INSTRUCT
;ONE ON THE USAGE OF FLOATING POINT NUMBERS AND MNEMONICS IN A PROGRAM.

FOUR    EQU     4.Ø                     ;EQUATE A FLOATING POINT REAL 4
NO_a    EQU     1.Ø                     ;EQUATE A FLOATING POINT REAL 1
NO_b    EQU     3.Ø                     ;EQUATE A FLOATING POINT REAL 3
NO_c    EQU     2.Ø                     ;EQUATE A FLOATING POINT REAL 2

;THE FOLLOWING IS THE DATA AREA WHERE THE RESULT WILL BE STORED

DATA SEGMENT WORD PUBLIC 'DATA'
FP4             DD      FOUR            ;DEFINE A SINGLE PRECISION 4
FP2             DD      NO_c            ;DEFINE A SINGLE PRECISION 2
FPa             DD      NO_a            ;DEFINE A SINGLE PRECISION "a"
FPb             DD      NO_b            ;DEFINE A SINGLE PRECISION "b"
FPc             DD      NO_c            ;DEFINE A SINGLE PRECISION "c"
CONTROL_WORD    DW      1 DUP (?)       ;DATA STORAGE FOR CONTROL_WORD
RESULTA         DD      1 DUP (?)       ;DATA STORAGE FOR RESULT X = -b + K/N
RESULTB         DD      1 DUP (?)       ;DATA STORAGE FOR RESULT X = -b + K/N
DATA    ENDS

;THE FOLLOWING IS THE CODE WHICH WILL PERFORM THE ROOT OF THE QUADRATIC
;EQUATION.

CODE SEGMENT WORD PUBLIC 'CODE'
;ASSUME THE DS REGISTER IS ADDRESSING THE "DATA SEGMENT".

        MOVE    CONTROL_WORD,Ø          ;LOAD THE CONTROL WORD

;BITS Ø THRU 5; INTERRUPTS AND EXCEPTION MASKS - NONE MASKED
;BIT 6:RESERVED - SET TO ZERO
;BIT 7:INTERRUPT ENABLE MASK - NOT MASKED
;BITS 8 AND 9: PRECISION CONTROL - SET TO 24 BITS (SINGLE PRECISION)
;BITS 1Ø AND 11: ROUNDING CONTROL - ROUND TO NEAREST OR EVEN
;BIT 12:INFINITY CONTROL - SET FOR PROJECTIVE
;BIT 13 THRU 15: RESERVED - SET TO ZERO

        FLDCW   CONTROL_WORD            ;LOAD THE CONTROL WORD INTO THE 8Ø87
        FLD     FPb                     ;LOAD THE NUMBER "b" (3)
        FCHS                            ;CHANGE THE SIGN OF b TO NEGATIVE.
```

```
:F1:SAMPLE.ASM  PAGE 2

;STEP 3:EXCHANGE "N" WITH 1st b @ ST(5)
;SOLVE FOR X- WHERE X = b - K/N

                FLD     FPb             ;LOAD ANOTHER b
                FLD     FP4             ;LOAD A 4.Ø
                FLD     FPc             ;LOAD THE NUMBER "c" (2)
                FLD     FPa             ;LOAD THE NUMBER "a" (1)
                FLD     FP2             ;LOAD A 2.Ø

;AT THIS POINT, THE STACK SHOULD APPEAR AS DEPICTED IN STEP 1 OF
;TABLE 3-3.

                FENI                    ;ENABLE THE INTERRUPTS

;SOLVE FOR N

                FMUL    ST,ST(1)        ;STEP 2:MULTIPLY THE TOP STACK
                                        ;       REGISTERS AND STORE THE
                                        ;       RESULT IN THE TOP AND DO NOT
                                        ;       POP THE STACK.
                FXCH    ST(5)           ;STEP 3:EXCHANGE "N" WITH 1st b @
                                        ;       ST(3)

;SOLVE FOR L

                FMUL    ST,ST(Ø)        ;STEP 4:SQUARE THE 2nd b
                FXCH    ST(3)           ;STEP 5:EXCHANGE "L" WITH 4 @ ST(3)

;SOLVE FOR M

                FMUL                    ;STEP 6A:MULTIPLY 4 TIMES a AND POP
                                        ;        THE STACK.  THE RESULT IS @
                                        ;        ST.
                FMUL                    ;STEP 6B:MULTIPLY RESULT IN 6A TIMES
                                        ;       c AND POP STACK.  THE
                                        ;       RESULT IS @ ST.

;SOLVE FOR K

                FSUBR                   ;STEP 7A:SUBTRACT M FROM L. POP THE
                                        ;       STACK.  RESULT IS @ ST.
                FSQRT                   ;STEP 7B:RESOLVE SQUARE ROOT VALUE OF
                                        ;       THE RESULT IN STEP 7A.
                FST     ST(3)           ;STEP 8:PUT A COPY OF K INTO ST(3).

;SOLVE FOR X - WHERE X = -b + K/N

                FADD    ST,ST(1)        ;STEP 9A:SOLVE FOR -b+K.
                FDIV    ST,ST(2)        ;STEP 9B:DIVIDE RESULT IN 9A BY N.
                                        ;       RESULT IS @ ST.
```

```
:F1:SAMPLE.ASM  PAGE 3

;STORE THE RESULT IN MEMORY

                FSTP    RESULTA         ;STEP 1Ø:STORE THE RESULT OF 9B @
                                        ;        RESULTA IN MEMORY AND POP
                                        ;        THE STACK.

;SOLVE FOR X - WHERE X = -b - K/N

                FSUB    ST,ST(2)        ;STEP 11A:SOLVE FOR -b-K.
                FDIV                    ;STEP 11B:DIVIDE RESULT IN 11A BY N.
                                        ;         POP THE STACK AND STORE IN
                                        ;         ST.

;STORE THE RESULT IN MEMORY

                FSTP    RESULTB         ;STEP 12:STORE THE RESULT OF 11B @
                                        ;        RESULTB AND POP THE STACK.

CODE    ENDS
        END
```

The answers to the computation reside at "RESULTA" and "RESULTB":

		S	EXPONENT	FRACTION (23 BITS)
RESULTA = -1 =		1	Ø1111111	ØØØØØØØØØØØØØØØØØØØØØØØB
RESULTB = -2 =		1	1ØØØØØØØ	ØØØØØØØØØØØØØØØØØØØØØØØB

***

CHAPTER 4
PRINCIPLES OF OPERATION

4.1 INTRODUCTION

This chapter provides a functional description of the iSBC 337A MULTIMODULE NDP board. The functional description describes the operation of the 8Ø87-2 Numeric Data Processor, which is the only IC on the board.

4.2 THE CPU/NDP RELATIONSHIP

The iSBC 337A board uses the 8Ø87-2 which performs as a co-processor to the 8Ø86 or 8Ø88 Central Processor Unit. This software transparent extension of the CPU improves system performance by off loading the arithmetic burdens on the 8Ø87-2.

The 8Ø87-2 shares the CPU's bus interface devices, requiring no additional circuitry. The 8Ø87-2 operates on the CPU's local bus as an alternate bus master with status lines common to both processors. This co-occupancy of the local bus and status lines permits sharing of the interface devices used by the master processor already provided on the base board. Figure 4-1 is a block diagram of the iSBC 337A MNDP.

The 8Ø87 tracks the instruction queue of the CPU by monitoring its status pins (QS1 and QSØ); any floating-point instructions are copied into its queue by tapping the data bus in parallel during the CPU fetch. When a floating-point instruction is fetched from the queue, the CPU will not execute it. When a non-memory referenced floating-point instruction is fetched, the CPU totally ignores it and proceeds on with its own activities while the 8Ø87-2 acts upon it. This provides an overlapping of processor operations which enhances the processing time of other software by the CPU.

When the floating-point instruction is memory referenced, the CPU will calculate the effective address (EA) using its addressing modes and will then perform a "dummy read". This is a normal read cycle for the CPU with the exception that the CPU will not accept the data into its execution unit. Instead the 8Ø87-2 will capture the address of the operand during T1 and the data during T3 (or TWait) ANDed with ready ((T3 or TWait)+READY). In this way the 8Ø87-2 utilizes the CPU's addressing capabilities. At this point the 8Ø87-2 can start execution unless more data is needed. If more data is needed, the 8Ø87-2 requests control of the local bus and status lines through the request/grant protocol.

In the request/grant protocol, the control of the local bus is maintained through the request/grant mechanism. The RQ/GTØ pin of the 8Ø87-2 is connected to the RQ/GT1 pin of the CPU.. When the 8Ø87-2 needs the local bus to conduct memory transfers, it requests the bus by pulling the RQ/GTØ pin low for one clock cycle. (Refer to Figure 4-2.) This action occurs only after the CPU has had a chance to perform its "dummy read". During the CPU's next T4 or T1 state, a pulse, one clock cycle wide, will be issued by the CPU on the RQ/GT1 line to indicate to the 8Ø87-2 that the CPU has allowed the local bus and status lines to float and that it will enter a "hold acknowledge" state at the next clock cycle. At the completion of the 8Ø87-2's use of the local bus and the status lines, the 8Ø87-2 will lower the RQ/GTØ line once more for one clock cycle to indicate to the CPU that the "hold request" is about to end and that the CPU can reclaim the local bus and status lines at the next clock cycle.

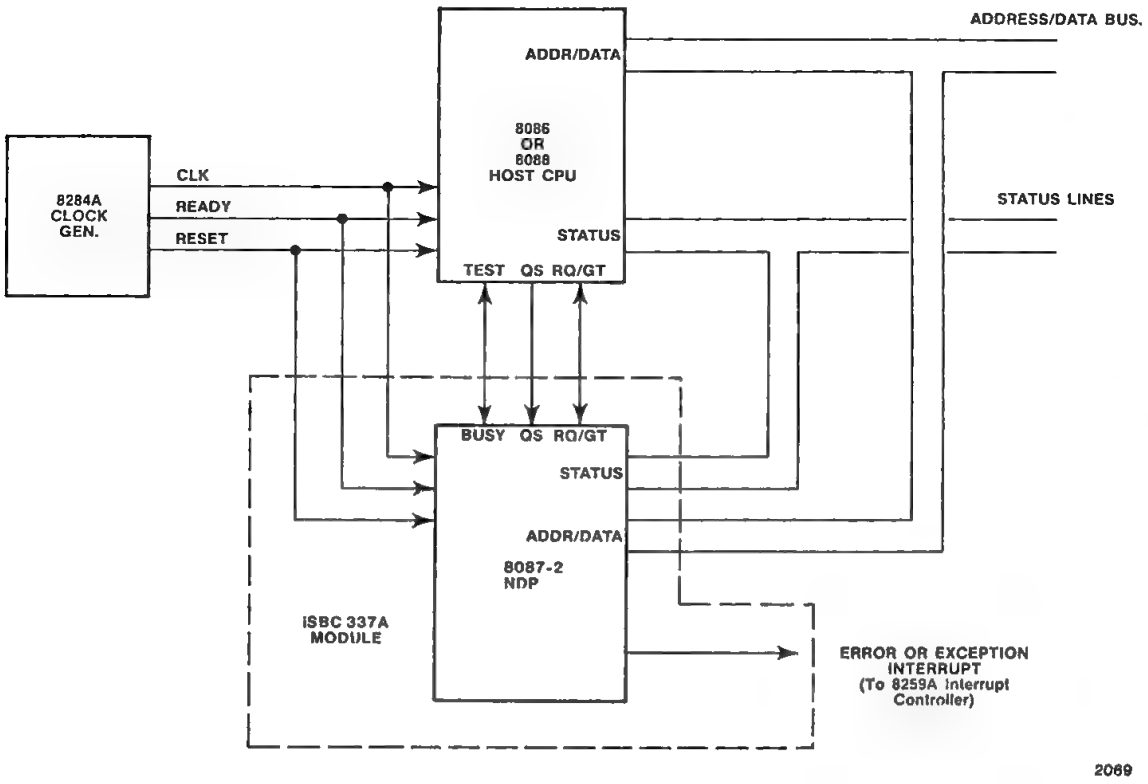
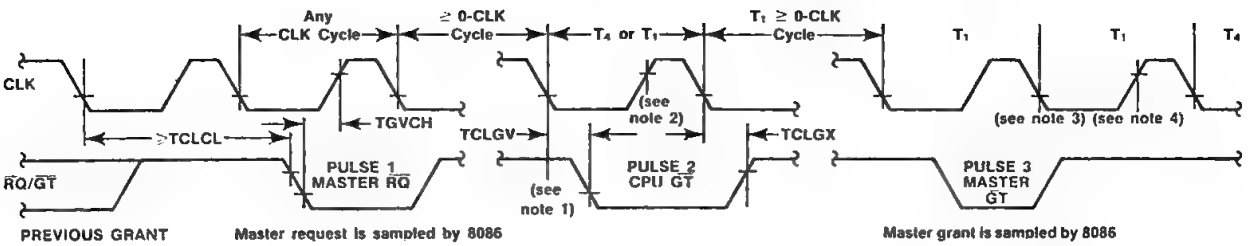


Figure 4-1. iSBC® 337A MNDP System Configuration



NOTES:
1. THE CPU FLOATS $\overline{S_2}$, $\overline{S_1}$, $\overline{S_0}$, FROM 1.1 STATE ON THIS EDGE
2. THE CPU FLOATS A_X, D_X, BUS, $\overline{BHE}$, AND $\overline{LOCK}$ ON THIS EDGE
3. THE OTHER MASTER FLOATS $\overline{S_2}$, $\overline{S_1}$, $\overline{S_0}$, FROM 1.1.1 STATE ON THIS EDGE
4. THE OTHER MASTER FLOATS A_X, D_X, $\overline{BUS}$, BHE, AND $\overline{LOCK}$ ON THIS EDGE

TCLCL — CLOCK CYCLE PERIOD OF CPU
TGVCH — RQ/GT SET UP 30 μSEC MIN
TCLGV — GRANT ACTIVE 85 μSEC MAX
TCLGX — GRANT INACTIVE 85 μSEC MAX

2073

Figure 4-2. Request/Grant Sequence Timing

The 8Ø87-2 will request the bus immediately after the "dummy read" when more data is required (typically for all load operations and memory priority instructions). In this case the request is made at the T4 state. When the data is to be written, the bus is requested as late as possible (at the T1 state). The RQ/GTØ signal from the CPU is returned to the base board via the P1 connector on the iSBC 337A board, for use on the base board. The RQ/GT1 signal from the CPU is also returned to the base board, but may not be used by the base board as the 8Ø87-2 uses it. The RQ/GT1 signal from the iSBC 337A board is routed to the base board via the P2 connector on the iSBC 337A board. This signal may be used for applications which would normally use the CPU's RQ/GT1 signal.

The interrupt output of the 8Ø87-2 is handled by routing the interrupt request down through the P2 connector of the iSBC 337A base board interface. The base board is responsible for manipulating and prioritizing the interrupt request and feeding it back to the CPU via the INT line. At that time, the CPU executes a software exception handling routine which is responsible for clearing the 8Ø87-2's busy status and servicing the exceptions in a proper manner.

The busy output pin of the 8Ø87-2 is connected to the test pin of the CPU.

A wait prefix is inserted, by the assembler, in front of each floating-point instruction. This will avoid overlapping of floating-point instructions. Prior to executing each floating-point instruction, the CPU will test the state of the 8Ø87-2's busy pin and will mark time until the 8Ø87-2's busy status and busy pin are reset indicating that the 8Ø87-2 is ready for another floating-point instruction. If an exception is encountered by the 8Ø87-2 and the interrupt is not masked off, an interrupt is issued and the busy status remains set until the CPU services the interrupt. If an interrupt occurs during the mark time, the CPU will return to the proper state in the

interrupt return sequence. If the interrupt output is masked off and an exception occurs, then a condition known as deadlock will exist where the busy status remains set and the CPU continues to wait. This condition is avoided by the 8Ø87-2 internally and explained in more detail in section 4.3.4.

4.2.1 BUS OPERATIONS

Each bus cycle consists of at least four clock cycles. These clock cycles are referred to as T1, T2, T3, and T4. The address is issued from the processor during T1 and data transfer occurs on the bus during T2 through T4. In the event that a ready indication is not received from the addressed device, wait states (TW) are inserted between T3 and T4. Each inserted wait state is of the same duration as a clock cycle.

The 8Ø87-2 sends status bits S2, S1, and SØ to provide type-of-cycle information to the 8288 Bus Controller. The 8288 Bus Controller generates the memory read and write commands, and issues control signals to the address latches and data transceivers. A multimaster system bus can be constructed with the use of the 8289 Bus Arbiter. The key bus arbiter inputs are the same as those for the 8288 Bus Controller (i.e., local status lines S2, S1, and SØ).

The 8Ø87-2 can operate on a 16-bit local bus (8Ø86 processor) or on an 8-bit local bus (8Ø88 processor). It detects the type of bus on the first cycle after power up by monitoring the status lines for a memory read and then checking the conditions of BHE/ for that first cycle. Since an 8Ø86 processor always fetches a word on the first cycle, its BHE/ will be low. Likewise, the 8Ø88 processor always does a byte fetch keeping that particular pin (called SSØ/ on the 8Ø88 processor) high. Therefore, if BHE/ is low on the first cycle, the 8Ø87-2 will use the 16-bit mode and will use the 8-bit mode if the BHE/ is high. When using an 8-bit local bus, ADØ-AD7 are bidirectional address/data lines and AD8-ADF become only bidirectional address lines.

4.2.2 READ OPERATION

The read cycle begins in T1 when the address is sent out on the bus and the Address Latch Enable (ALE) signal is asserted. The trailing edge of ALE locks the address into the 8282/8283 octal latch (or compatible TTL latch). The byte high enable (BHE) and address zero (ADØ) signals address the high, low, or both bytes. At T2 the address is removed from the local bus and the bus goes to a high impedance state. The memory read command signal (MRDC/) is also asserted at T2. This read command causes the addressed device to enable its data bus drivers to the system bus.

Sometime later, valid data will be available on the local bus and the addressed device will drive the ready line high. When the processor returns the read command to a high level, the addressed device will again place its data bus drivers in a high impedance state. If a transceiver (8286/8287) is required to buffer the local bus, the direction (DT/R) and enable (DEN) controls are provided by the 8288 Bus Controller.

4.2.3 WRITE OPERATION

A write cycle begins when the address is sent out on the bus and the Address Latch Enable (ALE) signal is asserted. The BHE/ and ADØ signals select the proper byte(s) of memory to be written. This data remains valid at least until the middle of T4. During T2, T3, and TW the advanced memory write command (AMWTC/) is asserted, while the normal memory write command (MWTC/) is asserted during T3 and TW only. The normal memory write command is used by older style memories that require valid data prior to the write command.

4.3 8087-2 ARCHITECTURE

As shown in Figure 4-3, the 8Ø87-2 is internally divided into two processing elements, the control unit (CU) and the numeric execution unit (NEU). The NEU executes all numeric instructions, while the CU fetches instructions, reads and writes memory operands, and executes the processor control type of instructions. The two elements are able to operate independently of one another, allowing the CPU to maintain synchronization with the CU while the NEU executes numeric instructions.

4.3.1 CONTROL UNIT

The CU keeps the 8Ø87-2 operating in synchronization with its host CPU. The 8Ø87-2 instructions are inter-mixed with the host CPU instructions in a single instruction stream that is fetched by the host CPU. By monitoring the status signals emitted by the host CPU, the 8Ø87-2 control unit can determine when an instruction is being fetched. When the instruction byte or word becomes available on the local bus, the 8Ø87-2 control unit taps the bus in parallel with the host CPU and obtains that portion of the instruction.

The CU maintains a queue of pre-fetched instructions identical to that in the 8Ø86/8Ø88 processor. By monitoring the CPU's queue status lines, the CU is able to obtain and decode instructions from the queue in synchronization with the CPU. In effect, both processors fetch and decode the instruction stream in parallel.

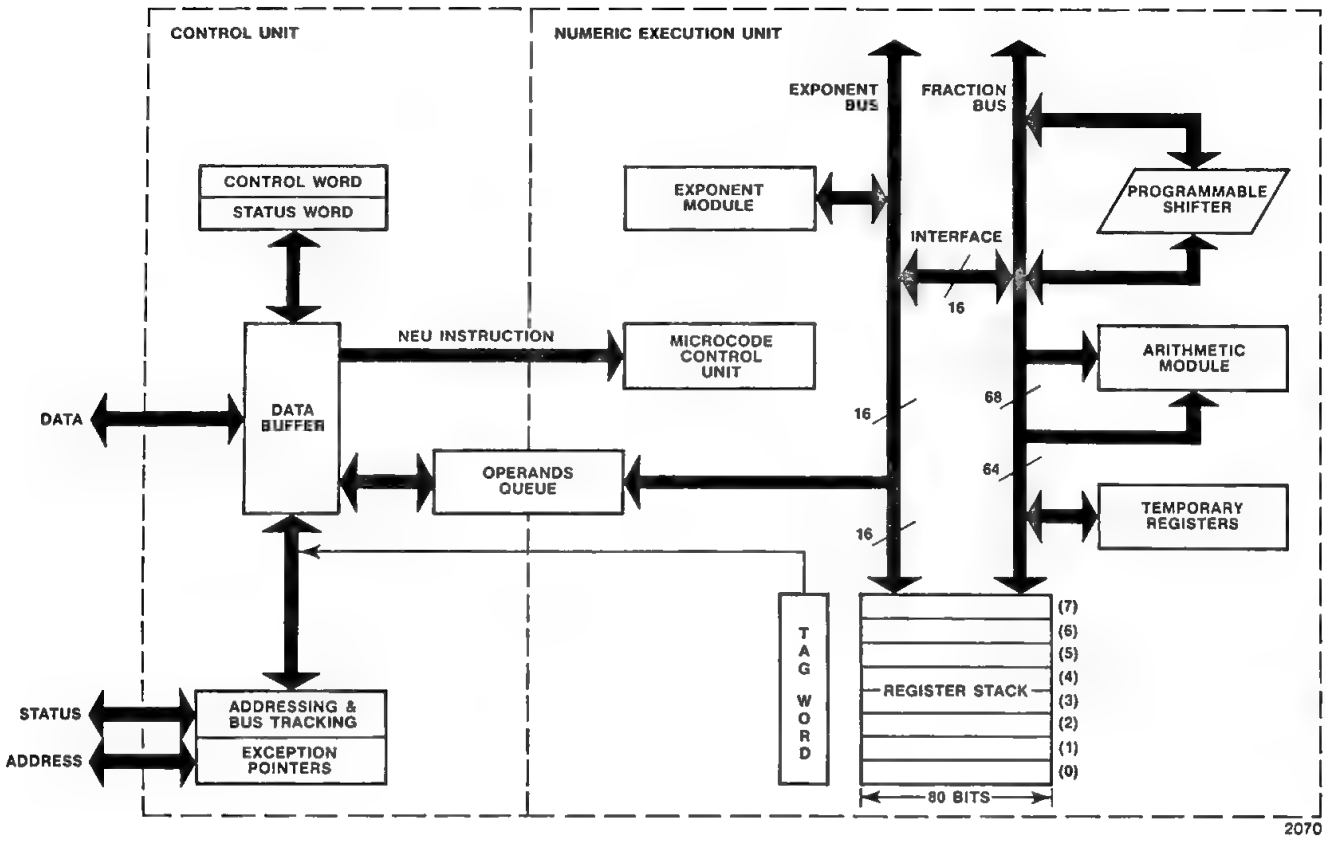


Figure 4-3. 8Ø87-2 Block Diagram

The two processors execute the instruction stream differently. The first five bits of all 8Ø87-2 machine instructions are identical. These five bits (the escape code) designate that they are iSBC 337A MNDP instructions. The CU ignores all instructions that do not match these bits, since these instructions are directed to the CPU only. When the CU decodes an instruction containing the escape code, it either executes the instruction itself, or passes it to the NEU, depending on the type of instruction.

The CPU distinguishes between escape code instructions that reference memory and those that do not. If the instruction refers to a memory operand, the CPU calculates the operand's address and then performs a read of the word at that location. This is a normal read cycle, except that the CPU ignores the data that is placed on the data bus. If the escape code instruction does not contain a memory reference, the CPU simply proceeds to the next instruction.

A given iSBC 337A MNDP instruction will require loading an operand from memory into the 8Ø87-2, or will require storing an operand from the 8Ø87-2 into memory, or will not reference memory at all. In the first two cases, the CU makes use of the read cycle initiated by the CPU. The CU captures and saves the operand address that the CPU places on the bus early in the read cycle. If the instruction is an 8Ø87-2 load, the CU additionally captures the first (and possibly the only) word of the operand when it becomes available on the bus. If the operand to be loaded is longer than one word, the CU immediately obtains the bus from the CPU and reads the rest of the operand in consecutive bus cycles. In a store operation, the CU captures and saves the operand address as in a load, and ignores the data word that follows. When the 8Ø87-2 is ready to perform the store, the CU obtains the bus from the CPU and writes the operand at the saved address using as many consecutive bus cycles as are necessary to store the operand.

4.3.2 NUMERIC EXECUTION UNIT

The NEU executes all instructions that involve the register stack. These instructions include arithmetic, comparison, transcendental, constant, and data transfer instructions. The data path in the NEU is 68 bits wide and allows internal operand transfers to be performed at very high speeds.

4.3.3 8087-2 INTERNAL FILE

The eight level internal file is treated as both a stack and a general register file. The TOP field in the 8Ø87-2 status word identifies the register which is the current stack top. After a reset or power up the TOP field will contain all zeros. A push to the stack will cause the TOP field to decrement to 111. The 2-bit tag field in the tag word identifies whether each register is empty or contains an operand. The format for data within the files is shown in figure 4-4.

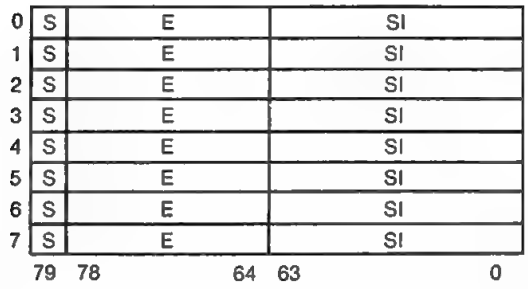
4.3.4 DEADLOCK

All of these conditions existing cause deadlock:

1.) The IEM bit is set, masking off any interrupts due to errors or exceptions.

2.) The CPU is executing an escape code instruction with a wait prefix.

3.) The 8Ø87-2 has issued an exception or an error.



Bit 79:
S = Sign of the significand. 1 represents negative and 0 represents positive.

Bits 64-78:
E = These 14-bits represent a biased exponent. The bias is $2^{14} - 1$ (16383). The range of the exponent is -16382 to +16383.

Bits 0-63:
M = 64-bit significand. Together with the sign bit, the significand represents a signed fraction in sign-magnitude notation. There is an implied binary point to the right of the most significant bit (bit 63) of the significand.

2071

Figure 4-4. 8Ø87-2 Internal File Format

The CPU is waiting for the busy pin to become inactive in order to execute the next escape code instruction. When the 8Ø87-2 issues an exception or an error, the B-bit remains set thereby keeping the busy pin in the active state. The busy state remains set until a clear error instruction has been executed. No interrupt is issued because the IEM bit is set, masking off interrupts. Therefore, the CPU is waiting for the 8Ø87-2 to deactivate its busy pin and the 8Ø87-2 is waiting for a clear error instruction in order to clear the busy condition. The 8Ø87-2 can detect a situation like this by continually monitoring the queue status and recognizing a wait prefix. When the IEM bit is set, a wait prefix has been detected, and an exception or error occurs, a deadlock signal is generated internally and logically ORed with the interrupt output circuitry to produce an interrupt to the CPU. Any user who chooses to run with the IEM bit set must understand that the CPU's interrupt must be enabled or an eternal deadlock condition can exist between the two processors.

***

CHAPTER 5
SERVICE INFORMATION

5.1 INTRODUCTION

This chapter provides a list of replaceable parts, service diagrams, and service and repair assistance instructions for the iSBC 337A MULTIMODULE NDP board.

5.2 REPLACEABLE PARTS

Table 5-1 provides a list of replaceable parts for the iSBC 337A board. Table 5-2 identifies and locates the manufacturers specified in the MFR CODE column in Figure 5-1. Intel parts that are available on the open market are listed in the MFR CODE column as COML. Every effort should be made to procure these parts from a local (commercial) distributor.

5.3 SERVICE DIAGRAMS

The parts location diagram and schematic diagram are provided in Figure's 5-1 and 5-2, respectively.

5.4 SERVICE AND REPAIR ASSISTANCE

United States customers can obtain service and repair assistance by contacting the Intel Product Service Marketing Administration in Phoenix, Arizona. Customers outside the United States should contact their sales source (Intel Sales Office or Authorized Distributor) for service information and repair assistance.

Before calling the Product Service Marketing Administration, you should have the following information available:

A. The date on which you received the product.

B. The complete model number (including dash number) and serial number for the product. These numbers are stamped onto the printed circuit boards.

C. Your shipping and billing addresses.

D. A purchase order number for billing purposes if your Intel product warranty has expired.

E. Extended warranty agreement information, if applicable.

Use the following numbers for contacting the Intel Product Service Marketing Administration group:

Regional Telephone Numbers

Western Region: 6Ø2-869-4951
Midwestern Region: 6Ø2-869-4392
Eastern Region: 6Ø2-869-4Ø45
International: 6Ø2-869-4862

TWX Number: 91Ø-951-133Ø, or 91Ø-951-Ø687

Always contact the Product Service Marketing Administration group before returning a product to Intel for service or repair. You will be given a "Repair Authorization Number", shipping instructions, and other important information which will help Intel provide you with fast, efficient service. If the product is being returned because of damage sustained during shipment from Intel, or if the product is out of warranty, a purchase order is necessary in order for the MCSD Technical Support Center to initiate the repair.

In preparing the product for shipment to Intel, use the original factory packaging material, if available. If the original packaging is not available, wrap the product in a cushioning material such as Air Cap TH-24Ø (or equivalent) manufactured by the Sealed Air Corporation, Hawthorne, N.J., and enclose in a heavy-duty corrugated shipping carton. Seal the carton securely, mark it "FRAGILE" to ensure careful handling, and ship it to the address specified by Product Service Marketing Administration group.

Table 5-1. Replaceable Parts

Reference Designation	Description	Mfr. Part No.	Mfr. Code	Qty.
C1,2	Cap., Ø.1 uF, Axial	CAM-Z59-7Ø5ØA	Cor.	2
U1	I.C., INTEL i8Ø87-2	1Ø8Ø71-ØØ1	Intel	1
P1	Pin, MULTIMODULE Sock.	CX49-79-1G1	Aug.	42
P1	Socket, Component Lead	76693-ØØ1	Berg.	4Ø

Table 5-2. List of Manufacturers' Codes

Mfr. Code	Manufacturer	Address
Aug.	Auget, Inc.	Attleboro, MA
Cor.	Corning Electronics	Corning, NY
Intel	Intel Corporation	Santa Clara, CA
Berg.	Berg Electronics	New Cumberland, PA

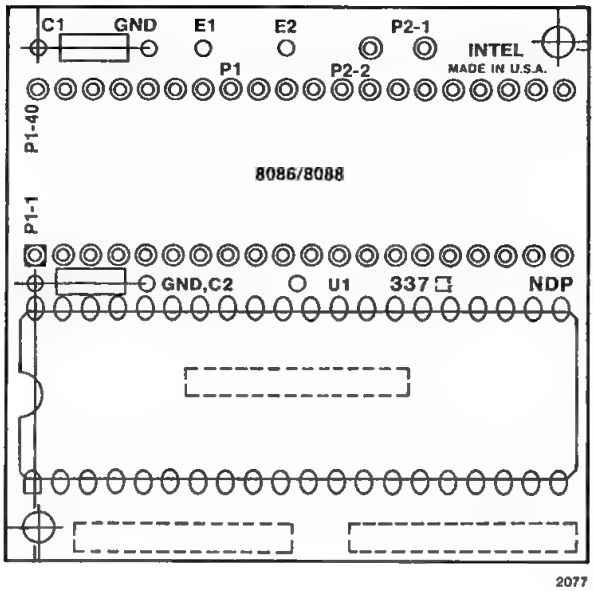


Figure 5-1. iSBC® 337A Board Parts Location Diagram

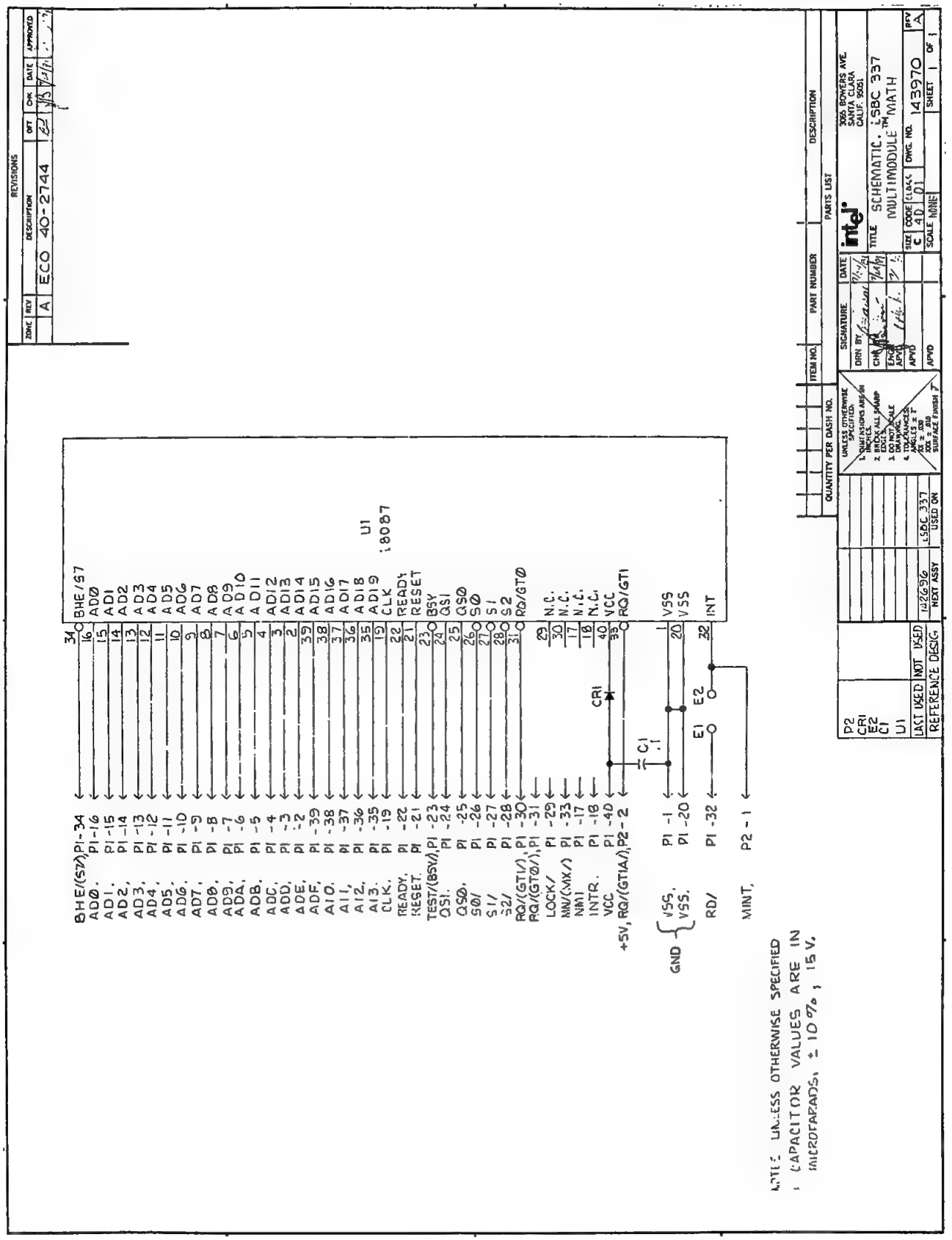


Figure 5-2. iSBC® 337A Board Schematic

***

INDEX

Primary references are underscored.

Architecture of the 8Ø87-2 4-5
    control unit 4-5
    deadlock 4-7
    internal file 4-7
    numeric execution unit 4-7

CPU/NDP relationship 4-1
    bus operations 4-4
    read operation 4-4
    request/grant protocol 4-2
    wait prefix insertion 4-3
    write operation 4-5

Data formats 3-7
    integer 3-1Ø
        long 3-1Ø
        short 3-1Ø
        word 3-1Ø
    packed binary coded decimal 3-11
    real 3-7
        double precision 3-8
        file precision 3-8
        single precision 3-7

Equipment supplied 1-1
Example problems 3-17
    example one program listing 3-17
    example two flowchart 3-2Ø
    problem two program detail 3-19
    example two program listing 3-22
    example two stack image 3-2Ø

Installation considerations 2-1
    connector configuration 2-2
    cooling requirements 2-2
    installation procedure 2-2
    jumper configuration 2-2
    physical dimensions 2-2
    power requirements 2-2

Programming 3-1
    8Ø87-2 instruction set 3-2
    escape code instructions 4-6
    instruction stream 4-6

Service information 5-1
    replaceable parts 5-1
    service diagrams 5-1
    service & repair assistance 5-1
        telephone numbers 5-2
Specifications 1-2
    337A board parts location diagram 5-4
    337A board schematic 5-5
    connectors P1 & P2 pin assignments 2-4
    connectors P1 & P2 signal functions 2-5
Status format 3-11
    control word 3-12
    status word 3-14
        condition codes 3-15
    tag word 3-17

***

REQUEST FOR READER'S COMMENTS

Intel's Technical Publications Departments attempt to provide publications that meet the needs of all Intel product users. This form lets you participate directly in the publication process. Your comments will help us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative. If you wish to order publications, contact the Intel Literature Department (see page ii of this manual).

1. Please describe any errors you found in this publication (include page number).

2. Does the publication cover the information you expected or required? Please make suggestions for improvement.

3. Is this the right type of publication for your needs? Is it at the right level? What other types of publications are needed?

4. Did you have any difficulty understanding descriptions or wording? Where?

5. Please rate this publication on a scale of 1 to 5 (5 being the best rating). _____

NAME _____ DATE _____

TITLE _____

COMPANY NAME/DEPARTMENT _____

ADDRESS _____

CITY _____ STATE _____ ZIP CODE _____
(COUNTRY)

Please check here if you require a written reply. ☐

WE'D LIKE YOUR COMMENTS . . .

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 79 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
5200 N.E. Elam Young Pkwy.
Hillsboro, Oregon 97123

OMO Technical Publications

intel®

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) **987-8080**

Printed in U.S.A.

iSBC®

0078/3K/1084/WCP